

Distributed Systems and Middleware Technologies  
JANET Home Simulator User Guide

Table of Contents of Contents

[JANET Simulator Commands Syntax 1](#_Toc85753512)

[SIMULATION START AND STOP 1](#_Toc85753513)

[SIMULATION MONITORING 1](#_Toc85753514)

[Print Database Tree 1](#_Toc85753515)

[Monitor Database Tree 1](#_Toc85753516)

[Print Running and Stopped Nodes 2](#_Toc85753517)

[Print JANET Simulator Mnesia Tables 2](#_Toc85753518)

[Print JANET Controller Mnesia Tables 2](#_Toc85753519)

[Remote Hosts Connectivity States 2](#_Toc85753520)

[JANET NODES START AND STOP 2](#_Toc85753521)

[Per-Node Start and Stop 2](#_Toc85753522)

[Per-Sublocation Start and Stop 2](#_Toc85753523)

[Per-Location Start and Stop 3](#_Toc85753524)

[All-Nodes Start and Stop 3](#_Toc85753525)

[DATABASE MANIPULATION 3](#_Toc85753526)

[Create 3](#_Toc85753527)

[Update 3](#_Toc85753528)

[Delete 3](#_Toc85753529)

[DATABASE BACKUP AND RESTORE 4](#_Toc85753530)

[Backup 4](#_Toc85753531)

[Restore 4](#_Toc85753532)

[Clear 4](#_Toc85753533)

JANET Simulator Commands Syntax

The complete set of commands offered by the JANET Simulator is presented in the tables below, with their parameters attuning to the following syntax:

* A lower-case parameter (i.e., an atom) should be left as it is:

jsim:print\_nodes(stopped) → jsim:print\_nodes(stopped)

* An upper-case parameter not enclosed in quotes denotes a non-string variable:

jsim:print\_ctr\_tree(Loc\_id) → jsim:print\_ctr\_tree(5)

* An upper-case parameter within quotes denotes a string variable:

jsim:print\_tree(user,”User”) → jsim:print\_tree(user,”Gianni”)

Some commands require the JANET Simulator application to be either running or stopped, and an in-depth description of each command along with its arguments and return values can be found in the *jsim* module (“apps/janet\_simulator/src/jsim.erl”).

The complete set of commands can also be printed directly in the *erl* shell via the jsim:help() command.

|  |  |  |
| --- | --- | --- |
| **SIMULATION START AND STOP** | | |
| **Command** | **Description** | **Must run** |
| jsim:run() | Starts the JANET Simulator application | **❌** |
| jsim:stop() | Stops the JANET Simulator application | **✔️** |
| jsim:shutdown() | Stops the JANET Simulator application and its ERTS | **–** |

|  |  |  |
| --- | --- | --- |
| **SIMULATION MONITORING** | | |
|  | | |
| Print Database Tree | | |
| **Command** | **Description** | **Must run** |
| jsim:print\_tree() | Prints the database contents indented as a tree | **–** |
| jsim:print\_tree(user, “User”) | Prints the database contents of a specific user indented as a tree | **–** |
| jsim:print\_tree(loc,Loc\_id) | Prints the database contents of a specific location indented as a tree | **–** |
| jsim:print\_tree(sub,Subloc\_id) | Prints the database contents of a specific sublocation indented as a tree | **–** |
|  |  |  |
| Monitor Database Tree | | |
| **Command** | **Description** | **Must run** |
| jsim:monitor\_tree() | Prints the database contents indented as a tree every 10 seconds | **–** |
| jsim:monitor\_tree(PeriodSecs) | Prints the database contents indented as a tree every *PeriodSecs* seconds | **–** |
| jsim:demonitor\_tree() | Stops the periodic printing of the database contents indented as a tree | **–** |
|  | | |
| Print Running and Stopped Nodes | | |
| **Command** | **Description** | **Must run** |
| jsim:print\_nodes() | Prints a summary of running and stopped nodes | **✔️** |
| jsim:print\_nodes(State)  - State = running | stopped | Prints a summary of running OR stopped nodes | **✔️** |
|  | | |
| Print JANET Simulator Mnesia Tables | | |
| **Command** | **Description** | **Must run** |
| jsim:print\_table() | Prints the contents of all database tables | **–** |
| jsim:print\_table(SimTable)  - SimTable = loc | sub | dev |   sup | ctrmgr | devmgr | Prints the contents of a specific database table | **–** |
| jsim:print\_record(SimTable,ID)  - SimTable = loc | sub | dev |   sup | ctrmgr | devmgr  - ID = Loc\_id | Sub\_id | Dev\_id | Prints a specific table record | **–** |
|  | | |
| Print JANET Controller Mnesia Tables | | |
| **Command** | **Description** | **Must run** |
| jsim:print\_ctr\_tree(Loc\_id) | Prints the contents of a controller’s database indented as a tree | **✔️** |
| jsim:print\_ctr\_table(Loc\_id) | Prints all tables of a controller’s database | **✔️** |
| jsim:print\_ctr\_table(Loc\_id,  CtrTable)  - CtrTable = sub | dev | Prints a specific table of a controller’s database | **✔️** |
|  | | |
| Remote Hosts Connectivity States | | |
| **Command** | **Description** | **Must run** |
| jsim:print\_rem\_hosts\_states() | Prints a summary of the connectivity states of the remote hosts used in the application | **✔️** |

|  |  |  |
| --- | --- | --- |
| **JANET NODES START AND STOP** | | |
|  | | |
| Per-Node Start and Stop | | |
| **Command** | **Description** | **Must run** |
| jsim:stop\_node(NodeType,NodeID)  - NodeType = ctr | dev  - NodeID = Loc\_id | Dev\_id | Stops the controller or device node of the given NodeID | **✔️** |
| jsim:restart\_node(NodeType,NodeID)  - NodeType = ctr | dev  - NodeID = Loc\_id | Dev\_id | Restarts the controller or device node of the given NodeID | **✔️** |
|  |  |  |
| Per-Sublocation Start and Stop | | |
| **Command** | **Description** | **Must run** |
| jsim:stop\_subloc(Sub\_id)  - Sub\_id = {Loc\_id,Subloc\_id} | Stops all device nodes in the given sublocation | **✔️** |
| jsim:restart\_subloc(Sub\_id)  - Sub\_id = {Loc\_id,Subloc\_id} | Restarts all device nodes in the given sublocation | **✔️** |
|  | | |
|  | | |
| Per-Location Start and Stop | | |
| **Command** | **Description** | **Must run** |
| jsim:stop\_loc(Loc\_id) | Stops the controller and all device nodes in the given location | **✔️** |
| jsim:restart\_loc(Loc\_id) | Restarts the controller and all device nodes in the given location | **✔️** |
|  | | |
| All-Nodes Start and Stop | | |
| **Command** | **Description** | **Must run** |
| jsim:stop\_all\_nodes() | Stops all controller and device nodes in the application | **✔️** |
| jsim:restart\_all\_nodes() | Restarts all controller and device nodes in the application | **✔️** |

|  |  |  |
| --- | --- | --- |
| **DATABASE MANIPULATION** | | |
| --- WARNING: Using these commands WILL lead to inconsistencies with the remote database --- | | |
| Create | | |
| **Command** | **Description** | **Must run** |
| jsim:add\_location(Loc\_id, ”Name”,User,Port,”Hostname”) | Adds a new location, also starting its controller node if the application is running | **–** |
| jsim:add\_sublocation(Sub\_id,”Name”)  - Sub\_id = {Loc\_id,Subloc\_id} | Adds a new sublocation in a location | **–** |
| jsim:add\_device(Dev\_id, ”Name”,Sub\_id,Type,”HostName”)  - Sub\_id = {Loc\_id,Subloc\_id}  - Type = fan | light | door |  thermostat | conditioner | Adds a new device in a sublocation, also starting its node if the application is running | **–** |
|  | | |
| Update | | |
| **Command** | **Description** | **Must run** |
| jsim:dev\_config\_change(Dev\_id,Config)  - DevConfig = #devcfg record | Changes a device’s configuration | **✔️** |
| jsim:update\_dev\_subloc(Dev\_id,Sub\_id)  - Sub\_id = {Loc\_id,Subloc\_id} | Changes a device’s sublocation within its location | **–** |
| jsim:update\_loc\_name(Loc\_id,”Name”) | Updates a location’s name | **–** |
| jsim:update\_subloc\_name(Sub\_id,”Name”)  - Sub\_id = {Loc\_id,Subloc\_id} | Updates a sublocation’s name | **–** |
| jsim:update\_dev\_name(Dev\_id,”Name”) | Updates a device’s name | **–** |
|  | | |
| Delete | | |
| **Command** | **Description** | **Must run** |
| jsim:delete\_location(Loc\_id) | Deletes a location, along with all its sublocations and devices | **–** |
| jsim:delete\_sublocation(Sub\_id)  - Sub\_id = {Loc\_id,Subloc\_id} | Deletes a sublocation, moving its devices in the default sublocation | **–** |
| jsim:delete\_device(Dev\_id) | Deletes a device | **–** |

|  |  |  |
| --- | --- | --- |
| **DATABASE BACKUP AND RESTORE** | | |
| --- WARNING: Using these commands WILL lead to inconsistencies with the remote database --- | | |
| Backup | | |
| **Command** | **Description** | **Must run** |
| jsim:backup() | Backs up the database contents to the “db/mnesia\_backup.db” file | **–** |
| jsim:backup(“FileName”) | Backs up the database contents to “Filename” under the “db” directory | **–** |
|  | | |
| Restore | | |
| **Command** | **Description** | **Must run** |
| jsim:restore() | Restores the database to the contents of the “db/mnesia\_backup.db” file | **❌** |
| jsim:restore(“FileName”) | Restores the database to the contents of “FileName” under the “db” directory | **❌** |
|  | | |
| Clear | | |
| **Command** | **Description** | **Must run** |
| jsim:clear() | Clears all database contents | **❌** |